



Программный комплекс Систэм Платформ

SePlatform.HMI.SetPoints 2.2

Руководство пользователя

Редакция
1. Предварительная

Соответствует версии ПО
2.2.0



© ООО «СИСТЭМ СОФТ», 2022-2023. Все права защищены.

Авторские права на данный документ принадлежат ООО «СИСТЭМ СОФТ». Копирование, перепечатка и публикация любой части или всего документа не допускается без письменного разрешения правообладателя.

Содержание

1. О продукте	4
2. Установка и удаление	5
3. Создание конфигурационного файла	7
4. Встраивание в проект	22
4.1. Подключение (обновление) модуля SePlatform.HMI.SetPoints	22
4.2. Добавление экземпляра SetPoints в проект	23
5. Запуск в виде самостоятельного приложения	28
6. Загрузка карт уставок	29
7. Чтение уставок из ПЛК	30
8. Редактирование значений уставок	31
9. Просмотр и изменение настроек уставок	33
10. Сохранение уставок в ПЛК	34
11. Сохранение в архив и просмотр архивных значений	35
История изменений	37
2.1	37
2.1.1	37
Изменения документации	38
Редакция 1	38
Редакция 2	38
2.0	38
2.0.1	39
Изменения документации	39
Редакция 1	39
1.2	39
1.2.1	39
1.2.2	40
1.2.3	40
Изменения документации	40
Редакция 1	40
Редакция 2	40

1. О продукте

SePlatform.HMI.SetPoints - приложение для просмотра и редактирования уставок параметров технологического процесса. SePlatform.HMI.SetPoints можно запустить как отдельное приложение, либо встроить его в проект автоматизации, построенный на SePlatform.HMI.

SePlatform.HMI.SetPoints можно встраивать в проекты, работающие на ОС Windows и ОС Linux и доступные через веб-интерфейс.

2. Установка и удаление

ОС Windows

Для установки или удаления SePlatform.HMI.SetPoints используйте дистрибутив `seplatform.hmi.setpoints-<lng>-<version>.<arch>.msi`.



ПРИМЕЧАНИЕ

В названии файла `<lng>` - это язык компонента, `<version>` - номер версии компонента, а `<arch>` - целевая процессорная архитектура.

ОС Linux

Для установки или удаления SePlatform.HMI.SetPoints используйте пакет `seplatform.hmi.setpoints-<lng>-<version>.<arch>.rpm` или `seplatform.hmi.setpoints-<lng>-<version>.<arch>.deb` в зависимости от выбранного пакетного менеджера ОС.



ПРИМЕЧАНИЕ

В названии пакета `<lng>` - это язык компонента, а `<version>` - номер версии компонента.



ОБРАТИТЕ ВНИМАНИЕ

Команды установки/удаления нужно запускать с правами суперпользователя **root**, находясь в папке с установочным пакетом.



ОБРАТИТЕ ВНИМАНИЕ

В командах удаления вместо полного имени пакета указывайте имя `seplatform.hmi.setpoints`.

Пакетный менеджер YUM

Установка

```
yum install seplatform.hmi.setpoints-<lng>-<version>.<arch>.rpm
```

Удаление

```
yum remove seplatform.hmi.setpoints
```

Пакетный менеджер RPM

Установка

```
rpm -i seplatform.hmi.setpoints-<lng>-<version>.<arch>.rpm
```

Удаление

```
rpm -e seplatform.hmi.setpoints
```

Пакетный менеджер APT

Установка

```
apt-get install seplatform.hmi.setpoints-<lng>-<version>.<arch>.deb
```

Удаление

```
apt-get remove seplatform.hmi.setpoints
```

Пакетный менеджер DPKG

Установка

```
dpkg -i seplatform.hmi.setpoints-<lng>-<version>.<arch>.deb
```

Удаление

```
dpkg -r seplatform.hmi.setpoints
```

3. Создание конфигурационного файла

Для того, чтобы работать с уставками в SePlatform.HMI.SetPoints, нужно прописать информацию о картах уставок в конфигурационном файле и загрузить этот файл в SePlatform.HMI.SetPoints. Информацию о картах уставок нужно добавлять в конфигурационный файл вручную в формате JSON.

В папке установки SePlatform.HMI.SetPoints C:\Program Files\SePlatform\SePlatform.HMI.SetPoints (OC Windows) или /opt/SePlatform/SePlatform.HMI.SetPoints (OC Linux) есть файл-пример example.spcfg, содержимое которого можно брать за основу при формировании конфигурационного файла. Ниже описаны правила заполнения файла. Старайтесь придерживаться их, чтобы ваш файл без ошибок загрузился в SePlatform.HMI.SetPoints.

Карты уставок

Карта уставок в SePlatform.HMI.SetPoints выглядит как таблица, в каждой строке которой содержится информация об отдельном параметре ([стр. 8](#)) и его уставках ([стр. 8](#)).

параметр и его уставки

№	Наименование	Идентификатор	Ед. изм.	Нижняя аварийная	Нижняя преду...	Нижняя техно...	Верхняя техно...	Верхняя преду...	Верхняя авари...
1	Давление на напоре 00SRP52CP001	SRP52CP001	МПа		21	48	17	13	12
2	Давление на напоре 00SRP52CP002	SRP52CP002	МПа						
3	Давление на напоре 00SRP98CP001	SRP98CP001	МПа						
4	Температура 00SRP90CT001 первого подшипника насоса	SRP90CT001	°C						
5	Температура 00SRP90CT002 второго подшипника насоса	SRP90CT002	°C						
6	Давление на напоре 00SRP98CP002	SRP98CP002	МПа						
	Температура 00SRP90CT003								

Добавляйте карты уставок в раздел Tables. Максимально можно добавить 20 карт уставок.

```
{
  "Id": "B4033877-C025-40C3-A555-65BA8F45B9BB", "Tables":
  [
    {
      "Name": "Санитарно-бытовой корпус",
      "Id": "Idtable1",
      "Type": "Table",
      "DataType": "float",
      "Format": "%.2f",
      "Columns": [{...}, {...}, ...],
      "RowTypes": [ {...}, {...}, ... ],
      "Rows": [{...}, {...}, ...]
    },
    {
      "Name": "Корпус №3",
```

```

    "Id": "Idtable2",
    "Type": "Table",
    "DataType": "float",
    "Columns": [{...}, {...}, ...],
    "RowTypes": [ {...}, {...}, ... ],
    "Rows": [{...}, {...}, ...]
  }
]
}

```

Чтобы добавить новую карту уставок, добавьте внутри раздела Tables новый блок с полями Name, Id, Type, DataType и разделами Columns, Rows, RowTypes.

Name	Имя карты уставок, отображаемое в дереве карт уставок SePlatform.HMI.SetPoints.
Id	Идентификатор карты уставок. Можно указать любое значение, но помните о том, что у каждой карты уставки должен быть уникальный идентификатор.
Type	Тип узла в дереве карт уставок SePlatform.HMI.SetPoints. Укажите значение «Table», если вы хотите создать одиночную карту уставок, а не папку, объединяющую несколько карт уставок в группу (стр. 7).
DataType	Тип данных всех уставок в этой карте уставок. Указанный здесь тип можно переопределить внутри параметра (стр. 8) или уставки (стр. 12).
Format	<p>Точность отображаемых значений всех уставок в этой карте уставок. Указанную здесь точность можно переопределить внутри параметра (стр. 8) или уставки (стр. 12).</p> <p>Если вы не укажете здесь значение и не переопределите точность внутри параметра или уставки, для всех уставок в этой карте будет выставлена максимально допустимая точность в пределах типа их значений:</p> <ul style="list-style-type: none"> ➤ «%.7g» - float; ➤ «%.15g» - double; ➤ «%11u» - uint4, uint8; ➤ «%11d» - int8; ➤ «%d» - int1, uint1, int2, uint2, int4.
Columns	Массив, содержащий список столбцов (стр. 7) карты уставок.
Rows	Массив, содержащий список параметров (стр. 8) карты уставок.
RowTypes	Массив, содержащий типы параметров (стр. 10), используемых в пределах конфигурационного файла (можно не указывать).

Параметры и уставки

Добавляйте параметры вместе с их уставками в раздел Rows. Максимальное количество параметров определяется по формуле: $P_{\max} = 3000 / \text{количество уставок}$, где P_{\max} - количество параметров.


```
{
  "Id": "B4033877-C025-40C3-A555-65BA8F45B9BB",
  "Tables":
  [
    {
      "Name": "Санитарно-бытовой корпус",
      "Id": "Idtable1",
      "Type": "Table",
      "DataType": "float",
      "Columns": [{...}, {...}, ...],
      "Rows": [{
        "Id": "SRP52CP001",
        "Name": "Давление на напоре 00SRP52CP001",
        "EUnit": "МПа",
        "Format": "%.3g",
        "DataType": "float",
        "Tag": "TestSignals.Device1",
        "LoLo": {...},
        "Lo": {...}
      }, {
        "Id": "00SRP52CP002",
        "Name": "Давление на напоре 00SRP52CP002",
        "Tag": "TestSignals.Device2",
        "HiHi": {...}
      }
    ]
  ]
}
```

Чтобы добавить новый параметр, добавьте внутри раздела Rows новый блок с полями, описывающими сам параметр: Id, Name, EUnit, Tag, Type, DataType и полями, содержащими описание уставок. Для удобства вы можете вынести повторяющиеся поля в отдельные блоки (в раздел RowTypes ([стр. 10](#))), а внутри раздела Row просто ссылаться на эти типы.

Id	Идентификатор параметра, отображаемый в столбце Идентификатор таблицы карты уставок SePlatform.HMI.SetPoints.
Name	Имя параметра, отображаемое в столбце Наименование таблицы карты уставок SePlatform.HMI.SetPoints.
EUnit	Единица измерения всех уставок этого параметра, отображаемая в таблице карты уставок SePlatform.HMI.SetPoints. Указанное здесь значение можно переопределить внутри каждой уставки (стр. 12).

Format	<p>Точность отображаемых значений всех уставок этого параметра. Указанную здесь точность можно переопределить внутри каждой уставки (стр. 12).</p> <p>Если вы не укажете здесь значение и не переопределите точность в столбце (стр. 14) или внутри уставки, для всех уставок этого параметра будет выставлена максимально допустимая точность в пределах типа их значений:</p> <ul style="list-style-type: none"> ➤ «%.7g» - float; ➤ «%.15g» - double; ➤ «%llu» - uint4, uint8; ➤ «%lld» - int8; ➤ «%d» - int1, uint1, int2, uint2, int4.
Tag	<p>Тег параметра, который будет добавляться к тэгу уставки при построении полного пути к сигналам чтения/записи уставки. Можно не указывать, тогда для каждой уставки нужно прописать полный путь к сигналам.</p>
Type	<p>Тип параметра. Если вы не подготовили для параметра типы (стр. 10), можете не указывать здесь значение.</p>
DataType	<p>Тип данных всех уставок этого параметра, отображаемый в таблице карты уставок SePlatform.HMI.SetPoints. Можно не указывать, тогда для всех уставок параметра будет использоваться тип данных, указанный в карте уставок. Если же для отдельных уставок определены собственные типы, будут использоваться именно они.</p>
Поля уставок	<p>На каждый параметр вы можете добавить до 20 уставок. Имена уставок могут быть любыми, но каждое имя должно совпадать с именем, указанным в поле Id для столбца (стр. 14), в который помещается значение этой уставки.</p>

Типы параметров

Иногда у разных параметров есть поля с одинаковыми значениями. В таком случае удобно описывать эти поля отдельно, за пределами раздела Rows (с описанием параметров ([стр. 8](#))), а в разделе Rows ссылаться на них. Ниже представлен пример создания типов параметров.



ПРИМЕР

У параметров Давление на напоре 00SRP52CP001 и Давление на напоре 00SRP52CP002 есть общая уставка HiHi, и для полей EUnit и DataType заданы одинаковые значения.

```
"Rows": [{
  "Id": "SRP52CP001",
  "Name": "Давление на напоре 00SRP52CP001",
  "Tag": "TestSignals.Device1",
  "DataType": "double",
  "EUnit": "МПа",
  "HiHi": {
    "ReadTag": "HiHiLimit",
    "WriteTag": "HiHiLimit.writeValue",
    "MinLimit": 1,
    "MaxLimit": 5
  }
}, {
  "Id": "SRP52CP002",
  "Name": "Давление на напоре 00SRP52CP002",
  "Tag": "TestSignals.Device2",
  "DataType": "double",
  "EUnit": "МПа",
  "HiHi": {
    "ReadTag": "HiHiLimit",
    "WriteTag": "HiHiLimit.writeValue",
    "MinLimit": 1,
    "MaxLimit": 5
  }
}]
```

Поместим все одинаковые поля в тип с именем AP6_PRESS. Чтобы добавить тип, создайте раздел **RowTypes** и добавьте внутри новый блок с полями Name (Имя типа), EUnit, DataType и полями уставки HiHi.

```
"RowTypes": [ {
  "Name": "AP6_PRESS",
  "DataType": "double",
  "EUnit": "МПа",
  "HiHi": {
    "ReadTag": "HiHiLimit",
    "WriteTag": "HiHiLimit.writeValue",
    "MinLimit": 1,
    "MaxLimit": 5
  }
}
],
"Rows": [{... }, {...}]
```

Теперь в полях параметров Давление на напоре 00SRP52CP001 и Давление на напоре 00SRP52CP002 нужно сослаться на созданный тип. Для этого в поле Type укажите имя созданного типа AP6_PRESS.



```
"Rows": [{
  "Id": "SRP52CP001",
  "Name": "Давление на напоре 00SRP52CP001",
  "Tag": "TestSignals.Device1",
  "Type": "AP6_PRESS"
}, {
  "Id": "SRP52CP002",
  "Name": "Давление на напоре 00SRP52CP002",
  "Tag": "TestSignals.Device2",
  "Type": "AP6_PRESS"
}]
```

Добавление уставок и связи с тегами сигналов

Добавляйте уставки для каждого параметра ([стр. 8](#)) внутри раздела Rows. Максимально можно добавить 20 уставок в одну карту уставок.

```
{
  "Id": "B4033877-C025-40C3-A555-65BA8F45B9BB",
  "Tables":
    [
      {
        "Name": "Санитарно-бытовой корпус",
        "Id": "Idtable1",
        "Type": "Table",
        "DataType": "float",
        "Columns": [{...}, {...}, ...],
        "Rows": [{
          "Id": "SRP52CP001",
          "Name": "Давление на напоре 00SRP52CP001",
          "EUnit": "МПа",
          "DataType": "float",
          "Tag": "TestSignals.Device1",
          "LoLo": {
            "ReadTag": "LoLoLimit",
            "WriteTag": "LoLoLimit.writeValue",
            "Format": "%.3g",
            "MinLimit": 0,
            "MaxLimit": 98,
            "DataType": "float"
          },
          "Lo": {
            "ReadTag": "LoLimit",
            "WriteTag": "LoLimit",
            "MinLimit": "sp:LoLo"
          }, {
            "Id": "00SRP52CP002",
            "Name": "Давление на напоре 00SRP52CP002",
```



```


    "Tag": "TestSignals.Device2",
    "HiHi": {
      "ReadTag": "HiHiLimit",
      "WriteTag": "HiHiLimit.writeValue",
      "MinLimit": 10,
      "MaxLimit": 50}
    }
  ]
}

```

Чтобы добавить новую уставку, добавьте новый блок после полей параметра (стр. 8) с именем уставки и её полями ReadTag, WriteTag, DataType, MinLimit и MaxLimit. Имя уставки может быть любым, но оно обязательно должно совпадать с именем, указанным в поле Id для столбца (стр. 14), в который помещается значение этой уставки.

Чтобы связать указанные уставки и сигналы сервера, укажите в поле ReadTag тег сигнала, из которого будет читаться значение уставки, а в поле WriteTag - тег сигнала, в который будет записываться значение уставки. Полный путь к сигналам строится с учетом тега, указанного в параметре (поле Tag), если он задан.

ReadTag	<p>Тег сигнала, из которого будет читаться значение уставки и помещаться в таблицу SePlatform.HMI.SetPoints. Вы можете указать полный тег, либо только его часть. Часть, общую для всех уставок параметра, можно вынести в поле Tag параметра.</p> <p>Если вы хотите указать, что в ПЛК (в архиве) нет такой уставки, оставьте это поле пустым.</p> <div>  ПРИМЕР <pre>"ReadTag": ""</pre> </div>
WriteTag	<p>Тег сигнала, в который будет записываться значение уставки из таблицы SePlatform.HMI.SetPoints. Вы можете указать полный тег, либо только его часть. Часть, общую для всех уставок параметра, можно вынести в поле Tag параметра.</p> <p>Если вы хотите указать, что уставка доступна только для чтения, оставьте это поле пустым.</p> <div>  ПРИМЕР <pre>"WriteTag": ""</pre> </div>

Format	<p>Точность отображаемого значения уставки. Указанную здесь точность можно переопределить внутри каждой уставки (стр. 12). Можно не указывать, тогда для уставки будет использоваться точность, указанная в карте уставок (стр. 7) или параметре (стр. 8).</p> <p>Если вы не укажете здесь значение и для уставки не определена точность выше (в карте уставок (стр. 7) или параметре (стр. 8)), для уставки будет выставлена максимально допустимая точность в пределах типа её значения:</p> <ul style="list-style-type: none"> ➤ «%.7g» - float; ➤ «%.15g» - double; ➤ «%llu» - uint4, uint8; ➤ «%lld» - int8; ➤ «%d» - int1, uint1, int2, uint2, int4.
MinLimit /MaxLimit	<p>Минимальное/максимальное значение уставки. Вы можете не указывать точное значение, а сослаться на другую уставку. Для этого укажите идентификатор другой уставки с префиксом «sp:».</p> <div style="border: 1px solid #0070c0; padding: 10px; margin: 10px 0;"> <p> ПРИМЕР</p> <pre>"MinLimit": "sp:LoLo"</pre> </div> <p>Можно не указывать, тогда допустимыми будут считаться границы значений для типа данных, указанного для параметра.</p>
DataType	<p>Тип данных уставки. Можно не указывать, тогда для уставки будет использоваться тип данных, указанный в карте уставок (стр. 7) или параметре (стр. 8).</p>

Столбцы карты уставок

Добавляйте столбцы в раздел Columns.

```
{
  "Id": "B4033877-C025-40C3-A555-65BA8F45B9BB",
  "Tables":
    [
      {
        "Name": "Санитарно-бытовой корпус",
        "Id": "Idtable1",
        "Type": "Table",
        "DataType": "float",
        "Columns": [{
          "Id": "Number",
          "Name": "№",
          "Width": "50"
        }, {
          "Id": "Name",
          "Name": "Наименование"
        }, {
          "Id": "Id",
          "Name": "Идентификатор",
```

```

        "Width": 150
      }, {
        "Id": "LoLo",
        "Name": "Нижняя аварийная",
        "Width": 150
        "Format": "%.3g",
      }, {
        "Id": "Lo",
        "Name": "Нижняя предупредительная"
      }, {
        "Id": "HiHi",
        "Name": "Верхняя аварийная"
      }
    ],
    "Rows": [{...}, {...}, ...]
  }
}

```

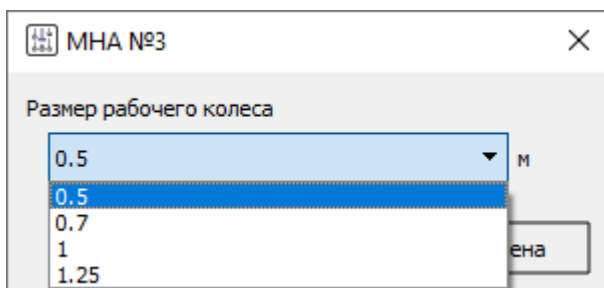
Все столбцы карты уставок заполняются данными из раздела Rows (с описанием параметров [\(стр. 8\)](#)). Первые четыре столбца: Number (имя - «№»), Id (имя - «Идентификатор»), Name (Имя - «Наименование») и EUnit (имя - «Единицы измерения») нужны для отображения информации о параметрах. Обязательно добавьте столбцы с идентификаторами Id и Name. Остальные столбцы содержат информацию об уставках, поэтому добавляйте их с учетом всех уставок [\(стр. 12\)](#), существующих в разделе Rows.

Чтобы добавить новый столбец, добавьте внутри раздела Columns новый блок с полями Id, Name, Width, EUnit, Format и Values.

Id	Идентификатор уставки (для первых четырех столбцов это идентификатор данных параметра). Для того, чтобы значения уставок помещались в нужный столбец, указывайте здесь именно то имя уставки (стр. 12) , которое вы указали в разделе Rows.
Name	Имя столбца, отображаемое в таблице карты уставок SePlatform.HMI.SetPoints.
Width	Ширина столбца в пикселях (можно не указывать).
Format	<p>Точность отображаемого значения уставок в этом столбце. Поле добавляется обычно для столбцов со значениями уставок. Указанную здесь точность можно переопределить внутри каждой уставки (стр. 12).</p> <p>Если вы не укажете здесь значение и не переопределите точность уставки в параметре (стр. 8) или внутри уставки (стр. 12), для всех уставок в этом столбце будет выставлена максимально допустимая точность в пределах типа их значений:</p> <ul style="list-style-type: none"> ➤ «%.7g» - float; ➤ «%.15g» - double; ➤ «%11u» - uint4, uint8; ➤ «%11d» - int8; ➤ «%d» - int1, uint1, int2, uint2, int4.
EUnit	Единица измерения, отображаемая при задании значения уставки. Можно не указывать, тогда для уставки будет использоваться единица измерения, указанная для параметра (стр. 8) .

Values

Список возможных значений уставки, отображаемый при задании значения уставки в SePlatform.HMI.SetPoints (можно не указывать).



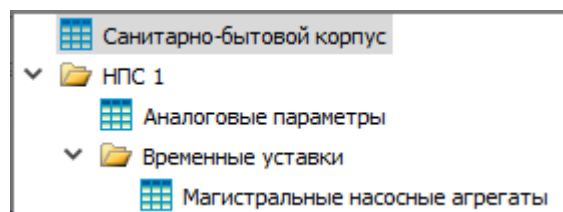
ПРИМЕР

Пусть уставка может принимать значения: «0.5», «0.7», «1» и «1.25». Чтобы сформировать выпадающий список, который будет отображаться в SePlatform.HMI.SetPoints при задании значения уставки, задайте здесь значение в формате:

```
"Values": [0.5, 0.7, 1, 1.25]
```

Объединение карт уставок в группы

Вы можете объединять карты уставок ([стр. 7](#)) в группы (по принадлежности к технологическим системам, объектам и т.д., которые в дереве карт уставок SePlatform.HMI.SetPoints будут выглядеть как папки).



Чтобы создать новую группу, добавьте в массив Tables структуру с типом "группа" (с полем "Type": "Group") со своим именем (поле Name) и своим идентификатором (поле Id). Внутри этой структуры добавьте массив Tables с картами уставок. Так вы можете вкладывать группы друг в друга.

```
{
  "Id": "B4033877-C025-40C3-A555-65BA8F45B9BB",
  "Tables": [{...}, {
    "Name": "НПС Сатарино",
    "Id": "IdGroup1",
    "Type": "Group",
    "Tables": [{...}, {...}]
  }]
}
```


Отключение уставок, наследуемых из типа параметра

Иногда уставки, описанные в типе параметра ([стр. 10](#)), могут не использоваться в каком-то параметре ([стр. 8](#)). Такие уставки можно отключить прямо внутри самого параметра. Для этого нужно явно прописать неиспользуемые уставки и для каждой добавить пустой тег чтения или тег записи (поля ReadTag или WriteTag).

```
"Rows": [{
  "Id": "TT111",
  "Name": "MHA1. Давление нефти на входе",
  "Tag": "NPS.MNA1.TTn11",
  "Type": "OIP",
  "Hi3": { "ReadTag": "" },
  "Hi2": { "ReadTag": "" },
  "Hi1": { "ReadTag": "" }
}, {
  "Id": "TT112",
  "Name": "MHA1. Давление нефти на выходе",
  "Tag": "NPS.MNA1.TTn12",
  "Type": "OIP",
  "Lo3": { "ReadTag": "" },
  "Lo2": { "ReadTag": "" },
  "Lo1": { "ReadTag": "" }
}
]
```

Доступ к настройкам уставки

Считывать настройки уставки с ПЛК (архива) ([стр. 33](#)) или вручную настраивать уставки ([стр. 33](#)) в SePlatform.HMI.SetPoints возможно, если в конфигурационном файле указать области памяти ПЛК, где хранятся настройки уставки, а также прописать теги сигналов для записи и чтения состояния уставки.

Чтобы добавить в конфигурационный файл сведения о настройках уставки, добавьте внутрь уставки блок Flags с полями Use, Sound и Message.

```
"Rows": [
  {
    "Id": "SRP52CP001",
    "Name": "Давление на напоре 00SRP52CP001",
    "EUnit": "МПа",
    "DataType": "float",
    "Tag": "TestSignals.Device1",
    "LoLo": {
      "ReadTag": "LoLoLimit",
      "WriteTag": "LoLoLimit.writeValue",
      "MinLimit": 0,
      "MaxLimit": 98,
      "DataType": "float",
      "Format": "%.1f",
      "Flags": {
```


```

        "Use": {
            "Bit": 0,
            "ReadTag": "LoLoLimit.Sets",
            "WriteTag": "LoLoLimit.Sets.writeValue"
        },
        "Sound": {
            "Bit": 1,
            "ReadTag": "LoLoLimit.Sets",
            "WriteTag": "LoLoLimit.Sets.writeValue"
        },
        "Message": {
            "Bit": 2,
            "ReadTag": "LoLoLimit.Sets",
            "WriteTag": "LoLoLimit.Sets.writeValue"
        }
    },
    "Lo": {
        "ReadTag": "LoLimit",
        "WriteTag": "LoLimit",
        "MinLimit": "sp:LoLo"
    }
}
]

```

Use	Группа полей описывающих настройки включения/отключения уставки.
Sound	Группа полей, описывающих настройки подачи звукового сигнала при срабатывании уставки.
Message	Группа полей, описывающих настройки выдачи сообщений при срабатывании уставки.

Поля внутри групп Use, Sound и Message:

Bit	<p>Номер бита в слове состояния из сигнала, указанного в поле ReadTag. Перед указанием значения выясните в какой области памяти контроллера хранится конфигурируемая вами настройка уставки и затем укажите нужный номер бита.</p> <p>Номер бита для настройки уставки должен быть указан обязательно, даже если поля ReadTag и WriteTag не заданы. Именно после указания номера бита в SePlatform.HMI.SetPoints появляется возможность считывания этой настройки с ПЛК (архива) (стр. 33) или изменения настроек уставки (стр. 33).</p> <div>  ПРИМЕЧАНИЕ При включении уставки/сообщения/звука в указанный бит записывается «true». </div>
-----	---

ReadTag

Тег сигнала, из которого будет читаться настройка уставки и помещаться в ячейку уставки в таблице SePlatform.HMI.SetPoints.



ОБРАТИТЕ ВНИМАНИЕ

Сигналы для хранения настроек уставки должны быть типа uint1/uint2/uint4/uint8.

Значение поля можно не указывать, тогда информация по конфигурируемой настройке не будет подгружаться с ПЛК (архива) ([стр. 33](#)) и отображаться в ячейке уставки.



ПРИМЕЧАНИЕ

Максимальное число тегов для чтения настроек уставок в одном конфигурационном файле - «1000». Тип данных тега ограничивает количество настроек, которое он может содержать. Например, если в теге используется тип int4, то в нем может храниться 32 настройки (4 байта * 8 бит). Максимально тег может хранить 64 настройки (в SePlatform.HMI.SetPoints считается, что максимальный размер слова состояния - 8 байт).

Максимальное число настроек в одном конфигурационном файле вычисляется по формуле:

$$64 \text{ (настройки)} * 1000 \text{ (тегов)} = 64\,000 \text{ (настроек)}$$

Помните про указанные ограничения при указании тегов с помощью поля ReadTag и номеров битов с помощью поля Bit.

Можно не указывать тег для конкретной настройки, а прописать его выше, в блоке Flags. Такой тег будет распространяться на все настройки уставки при условии, что для настроек указано значение поля Bit.



ПРИМЕР

```
{
  "LoLo": {
    "ReadTag": "LoLoLimit",
    "WriteTag": "LoLoLimit.writeValue",
    "MinLimit": 0,
    "MaxLimit": 98,
    "DataType": "float",
    "Format": "%.1f",
    "Flags": {
      "ReadTag": "LoLoLimit.Sets",
      "WriteTag": "LoLoLimit.Sets.writeValue",
      "Use": {
        "Bit": 0,
      },
      "Sound": {
        "Bit": 1,
      },
      "Message": {
        "Bit": 2,
      }
    }
  }
}
```

WriteTag

Тег сигнала, в который будет записываться настройка уставки, сделанная вручную в SePlatform.HMI.SetPoints.

**ОБРАТИТЕ ВНИМАНИЕ**

Сигналы для хранения настроек уставки должны быть типа uint1/uint2/uint4/uint8.

Значение поля можно не указывать, тогда конфигурируемую настройку нельзя будет изменять ([стр. 33](#)) (команда изменения настройки в ячейке уставки будет неактивна). Если не добавлять это поле ни в одну из настроек уставки (группы Use, Sound и Message, то настройки этой уставки в ячейке SePlatform.HMI.SetPoints будут доступны только для чтения.

**ПРИМЕЧАНИЕ**

Максимальное число тегов для записи настроек уставок в одном конфигурационном файле - «1000». Тип данных тега ограничивает количество настроек, которое он может содержать. Например, если в теге используется тип int4, то в нем может храниться 32 настройки (4 байта * 8 бит). Максимально тег может хранить 64 настройки (в SePlatform.HMI.SetPoints считается, что максимальный размер слова состояния - 8 байт).

Максимальное число настроек в одном конфигурационном файле вычисляется по формуле:

$$64 \text{ (настройки)} * 1000 \text{ (тегов)} = 64\ 000 \text{ (настроек)}$$

Помните про указанные ограничения при указании тегов с помощью поля WriteTag и номеров битов с помощью поля Bit.

Можно не указывать тег для конкретной настройки, а прописать его выше, в блоке Flags. Такой тег будет распространяться на все настройки уставки (см. пример для ReadTag).

Папка с оперативным архивом значений уставок

По умолчанию для оперативного архива используется путь:

C:\ProgramData\SePlatform\HMI.SetPoints\Archive. Если же у вас есть ограничения на запись файлов в папку по умолчанию, пропишите свой путь в разделе Archive (поле Path).

```
{
  "Id": "B4033877-C025-40C3-A555-65BA8F45B9BB",
  "Archive": {
    "Path": "C:\\WORK\\SetPoints\\Archive"
  },
  "Tables":
  [{...},{...}]
}
```

4. Встраивание в проект

Приложение SePlatform.HMI.SetPoints разработано средствами SePlatform.HMI и выполнено в виде проекта SePlatform.HMI. В собственные прикладные проекты SePlatform.HMI.SetPoints встраивается как отдельная библиотека - внешний модуль (подробнее о том, что такое внешний модуль, см. в руководстве пользователя на SePlatform.HMI).

Чтобы встроить SePlatform.HMI.SetPoints в проект и начать работу с приложением:

1. Подключите SePlatform.HMI.SetPoints к проекту как внешний модуль ([стр. 22](#)).
2. Добавьте экземпляр типа **SetPoints** в проект ([стр. 23](#)).
3. Укажите режим работы экземпляра типа **SetPoints** ([стр. 24](#)).



ПРИМЕЧАНИЕ

Если вы выбрали режим ПЛК, укажите также источник данных ([стр. 25](#)).

4. Можете указать файл с конфигурацией ([стр. 26](#)), если хотите, чтобы карты уставок загружались сразу же после запуска экземпляра типа **SetPoints** в режим исполнения.

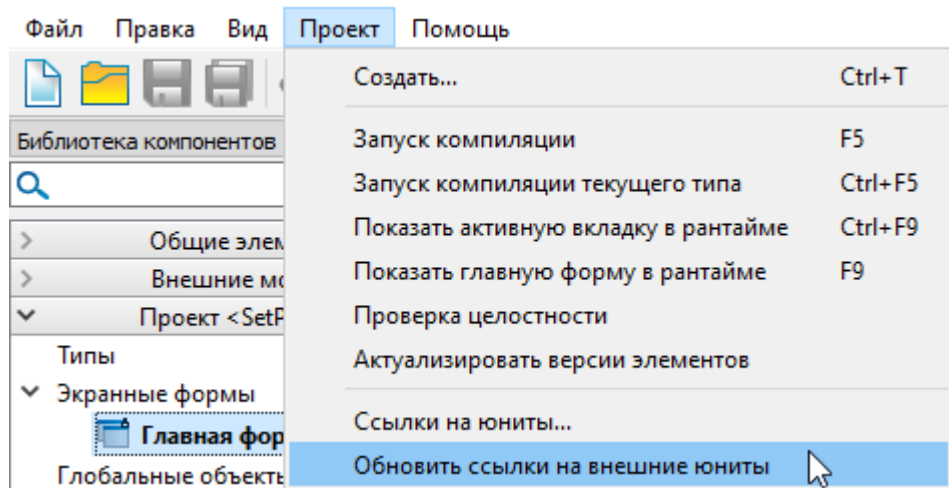
В интерфейсе встроенного экземпляра типа **SetPoints** вы увидите дерево карт уставок из указанного вами конфигурационного файла, сможете переключаться между картами и редактировать значения уставок, но не сможете пользоваться всеми остальными возможностями SePlatform.HMI.SetPoints. Чтобы расширить список доступных возможностей, используйте API, которое предоставляет экземпляр типа **SetPoints**.

4.1. Подключение (обновление) модуля SePlatform.HMI.SetPoints

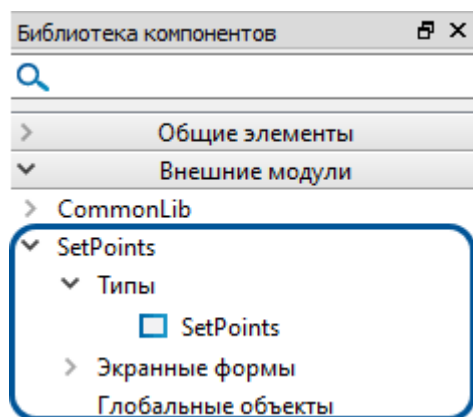
Подключить внешний модуль SePlatform.HMI.SetPoints к проекту

1. Создайте (если её там нет) в папке своего проекта папку `externals`, в которой нужно размещать файлы всех подключаемых внешних модулей.
2. Перейдите к папке, в которую устанавливаются все приложения SePlatform.HMI:
 - `C:\Program Files\SePlatform\SePlatform.HMI.Extensions` (OC Windows);
 - `/opt/SePlatform/SePlatform.HMI.Extensions` (OC Linux).
3. В папке уже должна быть папка `SetPoints`, появившаяся после установки SePlatform.HMI.SetPoints. Скопируйте эту папку `SetPoints` в созданную вами папку `externals`.
4. Откройте свой проект в дизайнера SePlatform.HMI.

5. Перейдите в меню **Проект** и выберите команду **Обновить ссылки на внешние юниты**.



Так вы обновите список внешних модулей своего проекта и новый модуль **SetPoints** появится в библиотеке компонентов.



Обновить встроенный модуль SePlatform.HMI.SetPoints

Чтобы обновить уже встроенный в проект SePlatform.HMI.SetPoints:

1. Установите нужную версию SePlatform.HMI.SetPoints.
2. Обновите файлы внешнего модуля **SetPoints** в папке **externals** своего проекта. Для этого замените существующую папку **SetPoints** на папку **SetPoints** из папки установки SePlatform.HMI.SetPoints:
 - C:\Program Files\SePlatform\SePlatform.HMI.Extensions (OC Windows);
 - /opt/SePlatform/SePlatform.HMI.Extensions (OC Linux).

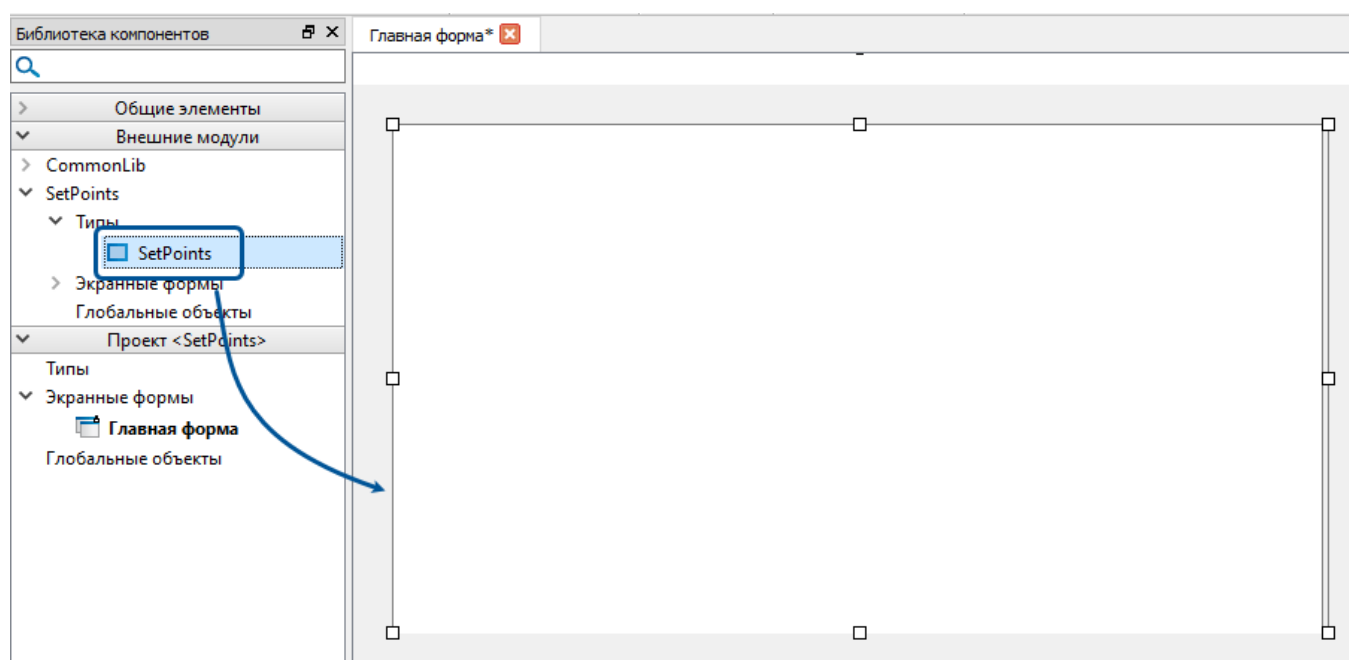
После обновления файлов модуля все экземпляры типа **SetPoints**, уже добавленные в проект, обновятся автоматически.

4.2. Добавление экземпляра SetPoints в проект

Добавить экземпляр типа SetPoints в проект

Основной элемент в составе подключенного внешнего модуля **SetPoints** - тип **SetPoints**.

Чтобы добавить экземпляр типа **SetPoints** в проект, просто перетяните тип на рабочую область.



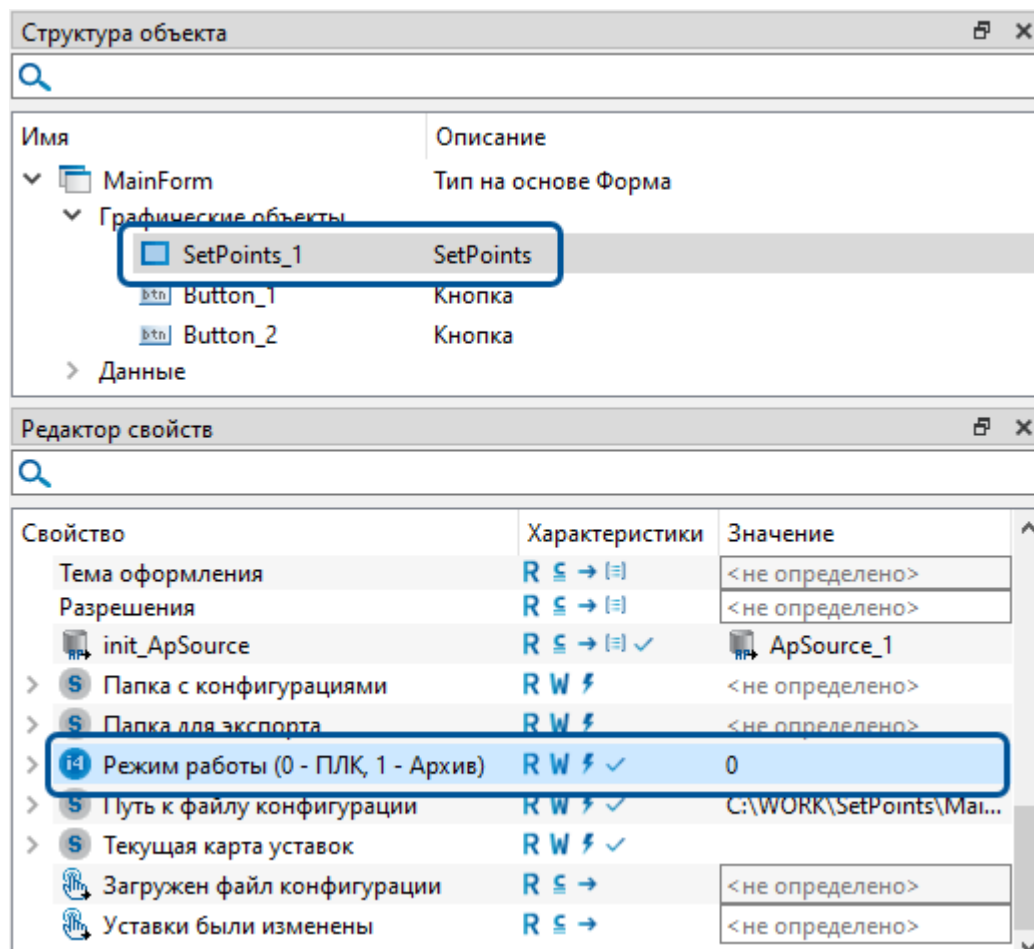
Вы можете добавить в проект сколько угодно экземпляров типа **SetPoints** и настроить каждый экземпляр под отдельные цели. Например, использовать разные экземпляры для просмотра значений уставок, полученных с ПЛК ([стр. 30](#)) и хранящихся в архиве ([стр. 35](#)).

Выбрать режим работы экземпляра типа SetPoints

С помощью SePlatform.HMI.SetPoints вы можете работать с данными, поступающими напрямую из ПЛК ([стр. 30](#)), либо хранящимися в архивных файлах ([стр. 35](#)). Чтобы переключаться между видами данных, выбирайте соответствующий режим работы экземпляра типа **SetPoints**.

Чтобы указать режим работы экземпляра типа **SetPoints**:

1. Перейдите к свойствам экземпляра (область **Редактор свойств**).
2. Найдите свойство **Режим работы (0 - ПЛК, 1 - Архив)** и укажите значение свойства.



ПРИМЕЧАНИЕ

Выбранный режим работы можно переключить в режиме исполнения с помощью средств API.

Подключить экземпляр типа SetPoints к источнику данных

Источник данных для SePlatform.HMI.SetPoints в режиме ПЛК ([стр. 30](#)) - сервер, передающий значения по протоколу TCP. Таким сервером может быть, к примеру, SePlatform.Data Server. Для настройки подключения к серверу в SePlatform.HMI используется компонент **Источник АР**. Поэтому для подключения SePlatform.HMI.SetPoints к серверу нужно настроить связь с каким-то из экземпляров типа **Источник АР**, существующих в проекте.

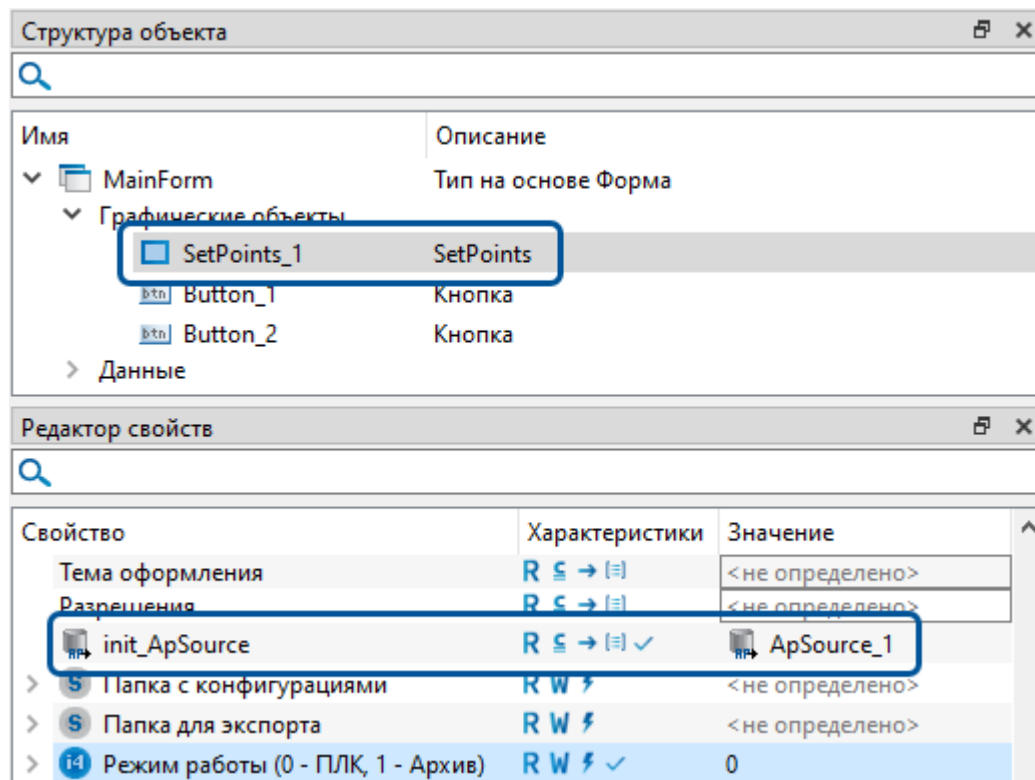


ОБРАТИТЕ ВНИМАНИЕ

Подключение к источнику данных нужно только если вы выгружаете значения уставок непосредственно из ПЛК ([стр. 30](#)). Если же вы просматриваете значения из архивного файла ([стр. 35](#)), подключаться к источнику необязательно.

Чтобы подключить экземпляр типа **SetPoints** к источнику данных:

1. Перейдите к свойствам экземпляра (область **Редактор свойств**).
2. Найдите свойство **int_ApSource** и укажите выбранный вами экземпляр типа **Источник АР**.



ОБРАТИТЕ ВНИМАНИЕ

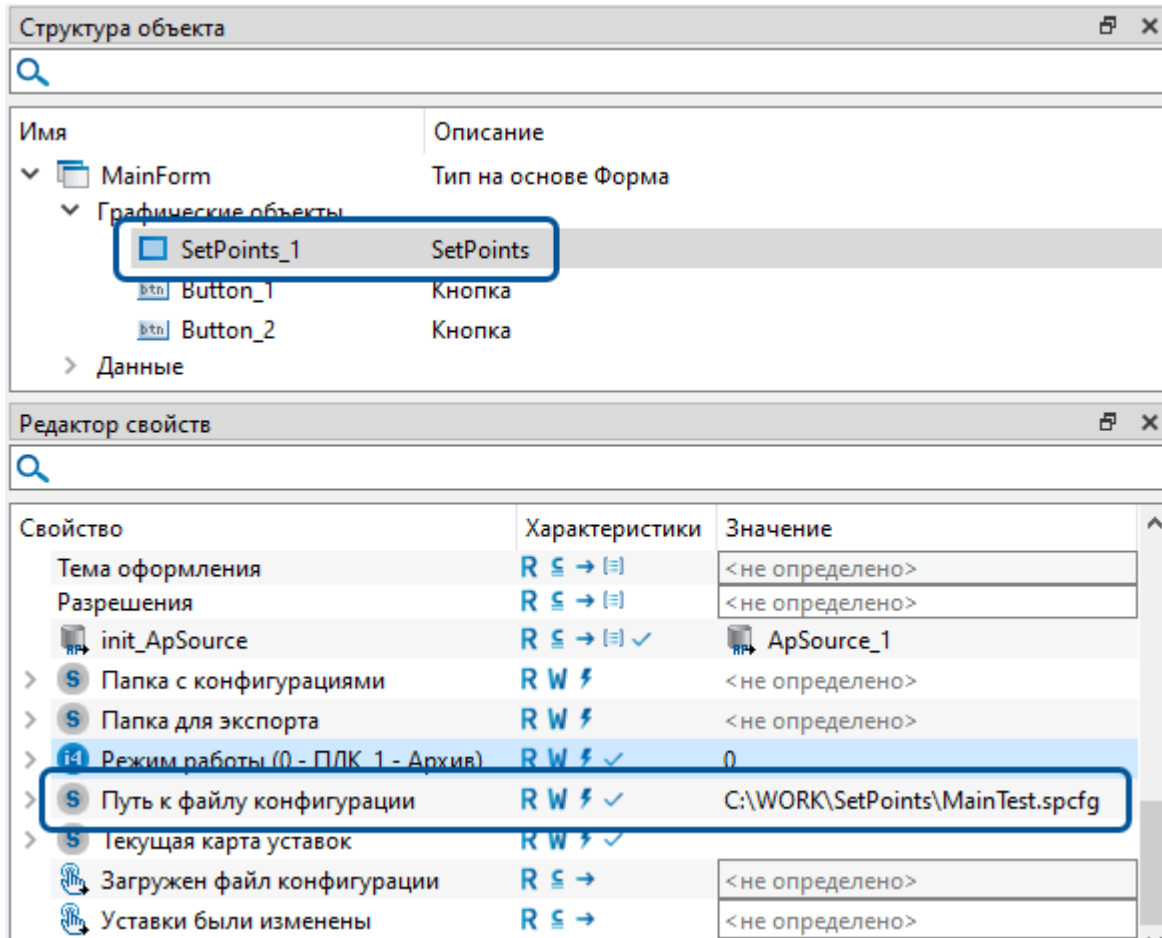
Если вам не удалось получить значения уставок с выбранного сервера, проверьте, что экземпляр типа **Источник АР** настроен верно:

- В свойстве **Хост** указан IP адрес или имя компьютера, на котором работает сервер.
- В свойстве **Активность** указано значение «true».

Указать файл конфигурации, загружаемой при запуске

Чтобы указать файл конфигурации, которая будет открываться в экземпляре типа **SetPoints** сразу же при запуске, выполните:

1. Перейдите к свойствам экземпляра (область **Редактор свойств**).
2. Найдите свойство **Путь к файлу конфигурации** (**ConfigFilePath**) и пропишите путь к файлу.



3. Перейдите к событиям формы, на которую вы добавили экземпляр типа **SetPoints**.
4. Найдите событие **Opened**. В его обработчике вызовите функцию **Open** экземпляра типа **SetPoints** и передайте ей путь к конфигурации, который вы указали в свойстве **Путь к файлу конфигурации** (**ConfigFilePath**).

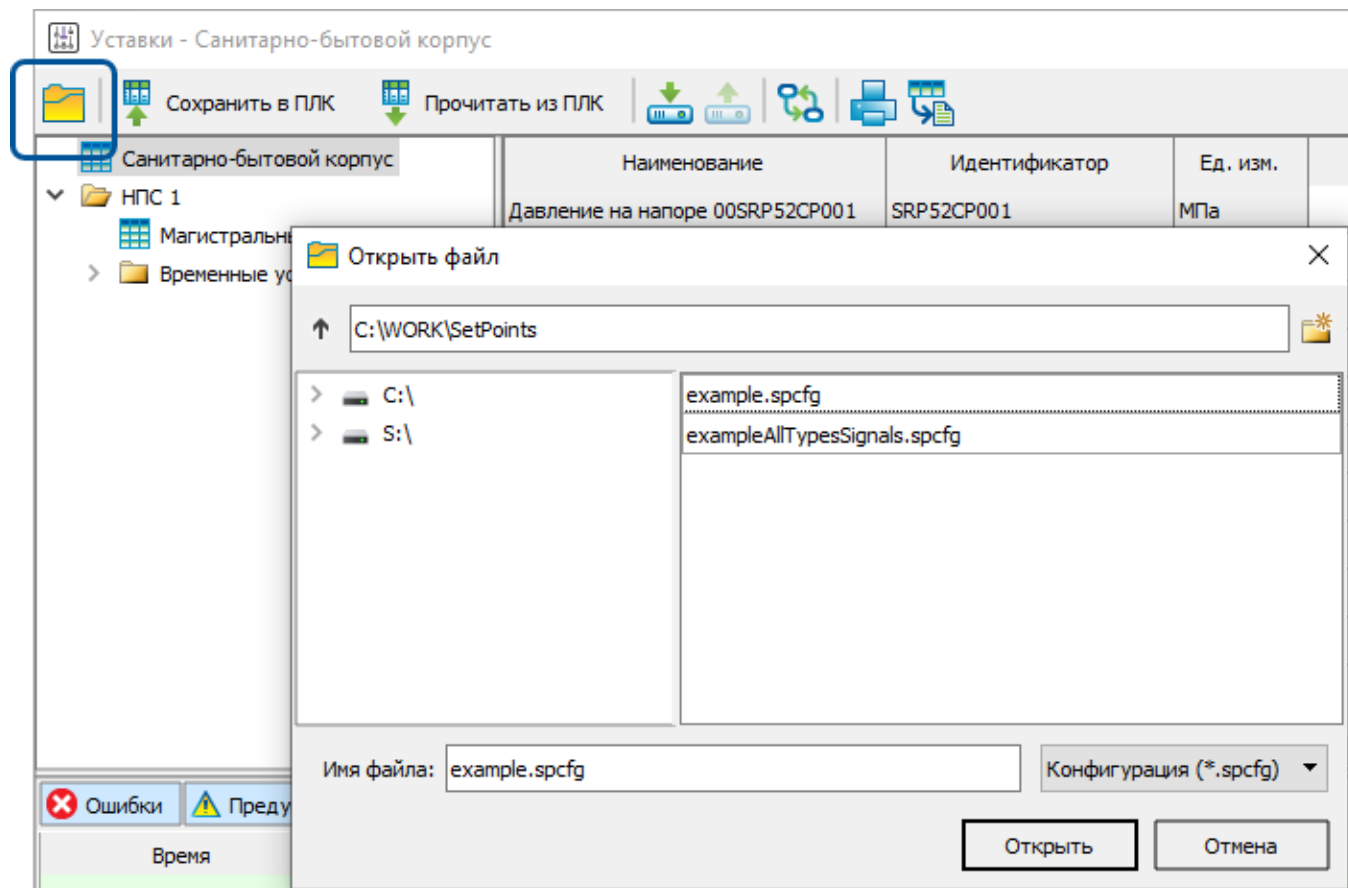
```
SetPoints_1.Open(SetPoints_1.ConfigFilePath);
```

5. Запуск в виде самостоятельного приложения

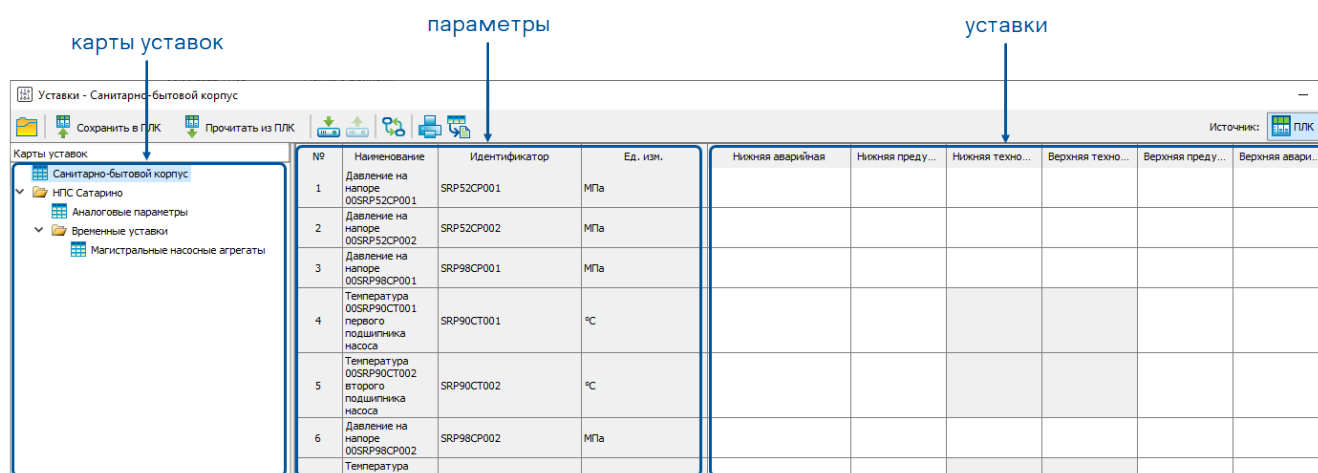
Чтобы открыть SePlatform.HMI.SetPoints как отдельное приложение, воспользуйтесь командой Пуск → SePlatform → SePlatform.HMI.SetPoints.

6. Загрузка карт уставок

Чтобы в SePlatform.HMI.SetPoints появился список карт уставок, нужно загрузить предварительно подготовленный конфигурационный файл. Нажмите кнопку **Открыть файл с конфигурацией** на панели инструментов и выберите файл.

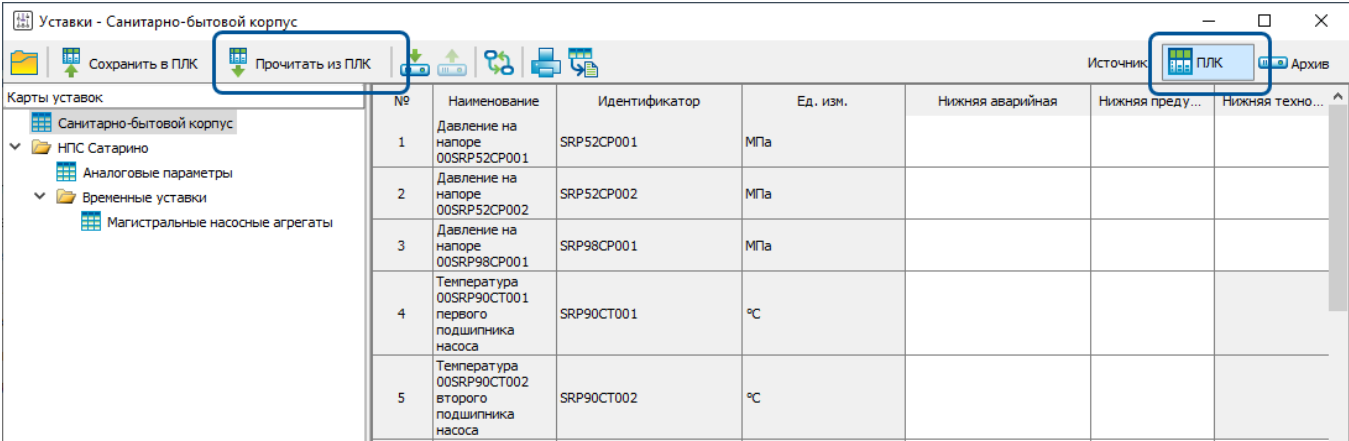


Загруженные карты уставок отображаются слева в виде дерева. Переключайтесь между картами, чтобы просматривать их содержимое: параметры и уставки.



7. Чтение уставок из ПЛК

Чтобы загрузить из контроллера значения уставок по всем картам уставок, переключитесь на источник ПЛК и нажмите кнопку **Прочитать из ПЛК** на панели инструментов.



ПРИМЕЧАНИЕ
При загрузке значений из контроллера все значения, которые вы ввели вручную, удаляются.

ПРИМЕЧАНИЕ
Некоторых уставок может не существовать в ПЛК (архиве). Чтобы ячейка несуществующей уставки отличалась от других ячеек в таблице SePlatform.HMI.SetPoints, отметьте эту уставку в конфигурационном файле. Ячейка несуществующей уставки окрашена серым цветом, ввести значение в такую ячейку нельзя.

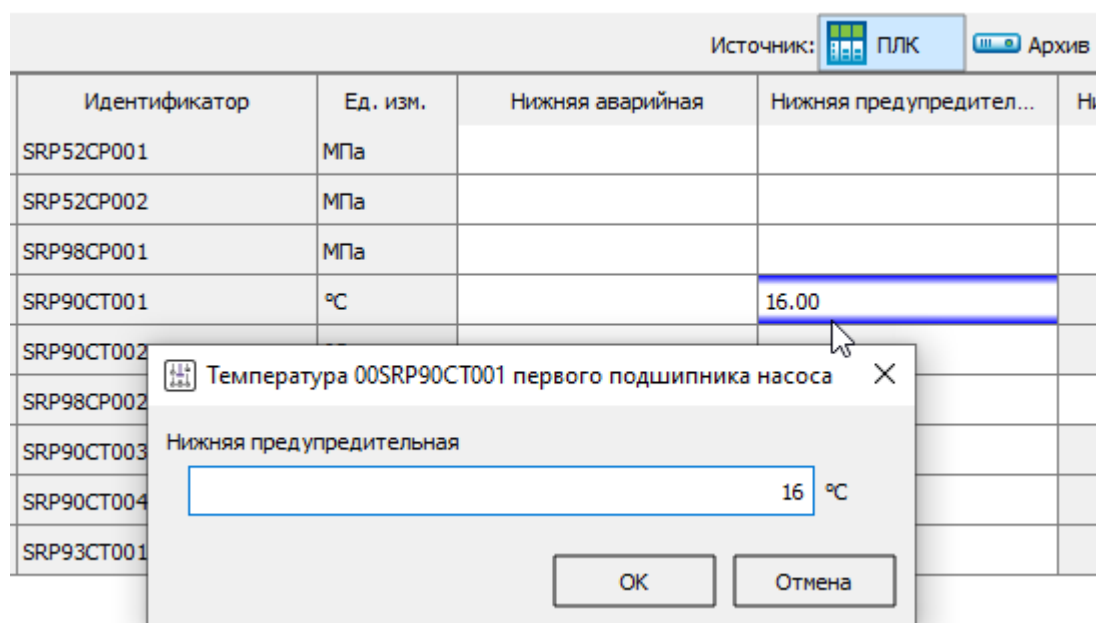
Ед. изм.	Нижняя аварийная	Нижняя предупредител...	Нижняя технологическая	Верхняя технологическая
МПа				
МПа				
МПа				
°C		16.00		
°C				
МПа				

Значения уставок в ПЛК могут выходить за ограничивающие пределы, заданные для уставок. Такие значения выделяются красным цветом.

№	Наименование	Идентификатор	Ед. изм.	Нижняя аварийная	Нижняя предуп...	Нижняя техно...	Верхняя техно...	Верхняя предуп...
1	Давление на напоре 00SRP52CP001	SRP52CP001	МПа		16	12	32	10
2	Давление на напоре 00SRP52CP002	SRP52CP002	МПа					
3	Давление на напоре 00SRP98CP001	SRP98CP001	МПа					

8. Редактирование значений уставок

Вы можете редактировать значения уставок, полученных с ПЛК или просматриваемых из архива. Чтобы редактировать значение уставки, щелкните два раза по значению и введите значение с учетом ограничивающих пределов, заданных для уставки.



ПРИМЕЧАНИЕ

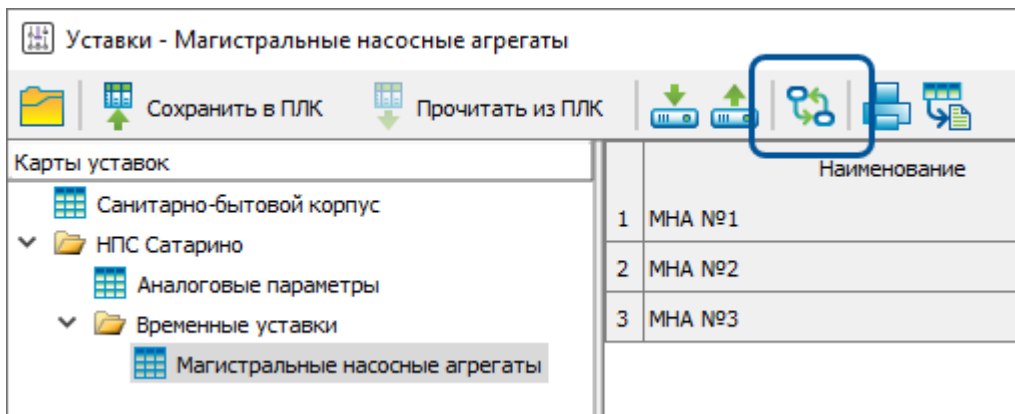
Чтобы запретить ввод значений для отдельной уставки в таблице SePlatform.HMI.SetPoints, отметьте эту уставку ([стр. 13](#)) в конфигурационном файле. Ячейка такой уставки окрашена серым цветом, а значение уставки можно только просматривать и сохранять в архив. Записать такую уставку в ПЛК нельзя.

№	Наименование	Идентификатор	Ед. изм.	Нижняя...	Нижняя предупредител...
1	Давление на напоре 00SRP52CP001	SRP52CP001	МПа		
2	Давление на напоре 00SRP52CP002	SRP52CP002	МПа		
3	Давление на напоре 00SRP98CP001	SRP98CP001	МПа		
4	Температура 00SRP90CT001 первого подшипника насоса	SRP90CT001	°C		16.00
5	Температура 00SRP90CT002 второго подшипника насоса	SRP90CT002	°C		
6	Давление на напоре 00SRP98CP002	SRP98CP002	МПа		

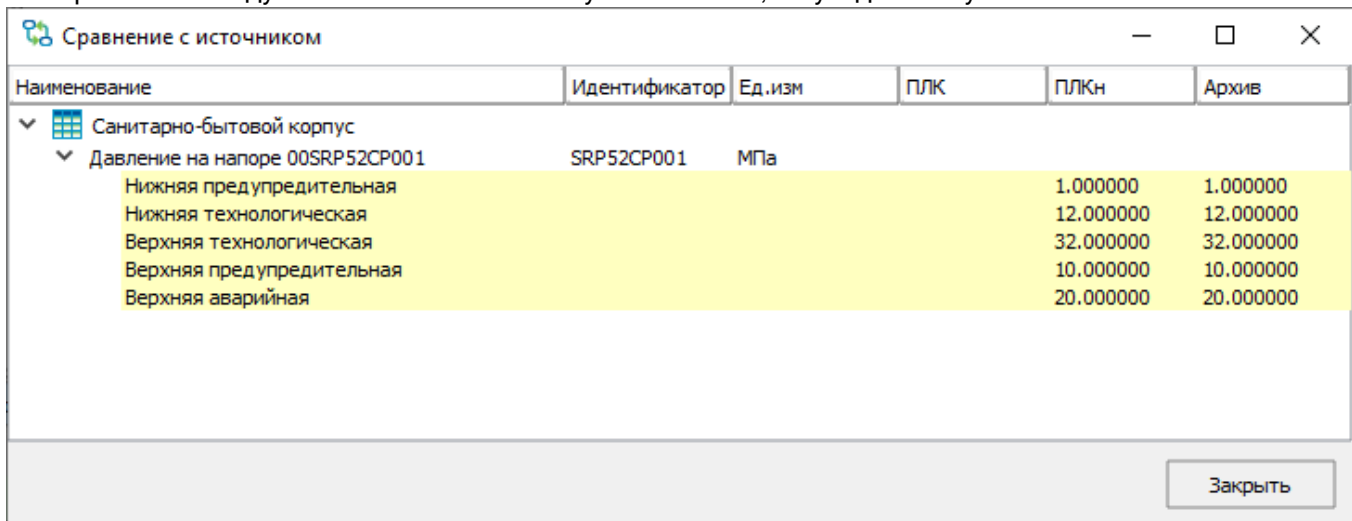
Если введенное значение отличается от значения в ПЛК, ячейка окрашивается в желтый цвет.

№	Наименование	Идентификатор	Ед. изм.	Нижняя аварийная	Нижняя преду...	Нижняя техно...	Верхняя техно...	Верхняя преду...
1	Давление на напоре 00SRP52CP001	SRP52CP001	МПа		1	12	32	10
2	Давление на напоре 00SRP52CP002	SRP52CP002	МПа					
3	Давление на напоре 00SRP98CP001	SRP98CP001	МПа					

Вы всегда можете проверить отличаются ли значения, отображаемые в таблице SePlatform.HMI.SetPoints от значений в ПЛК и архиве. Для этого нажмите кнопку **Сравнить с источником** на панели инструментов.



Если различия между значениями по каким-то уставкам есть, вы увидите эти уставки в списке.




9. Просмотр и изменение настроек уставок

На уровне ПЛК хранятся такие настройки уставки, как: включена или отключена уставка, разрешена или запрещена подача звукового сигнала, а также выдача сообщения при срабатывании уставки.

Считывание настроек уставок с ПЛК (архива)

Настройки каждой уставки, считываемые с ПЛК или архива, могут отображаться в ячейке рядом со значением уставки.

Уставки - Санитарно-бытовой корпус				
<div> <div>Сохранить в ПЛК</div> <div>Прочитать из ПЛК</div> <div></div> <div></div> <div></div> </div>				
Санитарно-бытовой корпус	Идентификатор	Ед. изм.	Нижняя аварийная	Нижняя предупредител...
> НПС Сатарино	SRP52CP001	МПа		
	SRP52CP002	МПа		
	SRP98CP001	МПа		
	SRP90CT001	°C		16 
	SRP90CT002	°C		



ОБРАТИТЕ ВНИМАНИЕ

Чтобы настройки уставки отображались в ячейке, добавьте описание настроек ([стр. 33](#)) в конфигурационный файл. Если вы хотите отключить видимость каких-то отдельных настроек ([стр. 21](#)) уставки, также используйте конфигурационный файл.

Изменение настроек уставок

Вы можете менять настройки уставок, полученных с ПЛК или просматриваемых из архива. Чтобы сменить настройки уставки, вызовите контекстное меню в ячейке и выберите нужную команду.

Верхняя технологическая	Верхняя предупредител...	Верхняя аварийная
	16	
	<div> <div>✓ Использовать уставку</div> <div>✓ Подавать звуковой сигнал</div> <div>✓ Выдавать сообщение</div> </div>	




ОБРАТИТЕ ВНИМАНИЕ

Чтобы команды изменения настроек уставки отображались в ячейке, добавьте описание настроек ([стр. 17](#)) в конфигурационный файл. Если вы хотите отключить возможность выбора каких-то отдельных команд ([стр. 21](#)), также используйте конфигурационный файл.

10. Сохранение уставок в ПЛК

Чтобы сохранить значения уставок по всем картам уставок, перейдите к нужному источнику (ПЛК или Архив) и нажмите кнопку **Сохранить** в ПЛК на панели инструментов.



ОБРАТИТЕ ВНИМАНИЕ

Будьте внимательны при загрузке значений в ПЛК. Помните, что ваши действия могут повлиять на технологический процесс.

Уставки - Санитарно-бытовой корпус

Сохранить в ПЛК

Прочитать из ПЛК

Источник: ПЛК

Архив

Карты уставок	Наименование	Идентификатор	Ед. изм.	Нижняя аварийная	Нижняя преду...	Нижняя техно...	Верхняя техно...	Верхняя преду...	Верхняя авари...
Санитарно-бытовой корпус	авление на входе DSRP52CP001	SRP52CP001	МПа		2	48	35	13	12
НПС Сатарино	авление на входе DSRP52CP002	SRP52CP002	МПа						
Временные уставки	авление на входе DSRP98CP001	SRP98CP001	МПа						
Магистральные насосные агрегаты									

11. Сохранение в архив и просмотр архивных значений

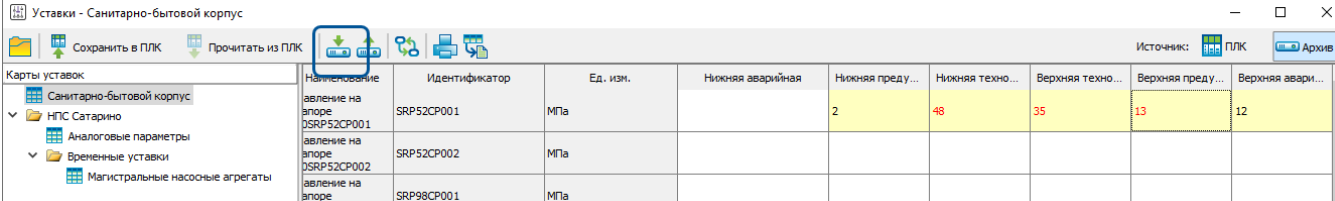
Вы можете периодически сохранять значения уставок по всем картам уставок в архивные файлы. Это удобно, когда с одной картой уставок работают разные сотрудники с разных рабочих мест. Сохраненные в архив файлы всегда можно открыть с другого компьютера. Ниже показан пример архивного файла.

```

1  {
2      "Idtable1": {
3          "SRP52CP001": {
4              "Lo": "1.000000",
5              "LoTech": "12.000000",
6              "HiTech": "32.000000",
7              "Hi": "10.000000",
8              "HiHi": "20.000000"
9          },
10         "SRP52CP002": {},
11         "SRP98CP001": {},
12         "SRP90CT001": {},
13         "SRP90CT002": {},
14         "SRP98CP002": {},
15         "SRP90CT003": {},
16         "SRP90CT004": {},
17         "SRP93CT001": {}
18     },
19     "Idtable_4_setpoints": {
20         "PT111": {},
21         "PT112": {},
22         "PT113": {}
23     },
24     "IdTable3MNS": {
25         "XT111": {},
26         "XT112": {},
27         "XT113": {}
28     }
29 }

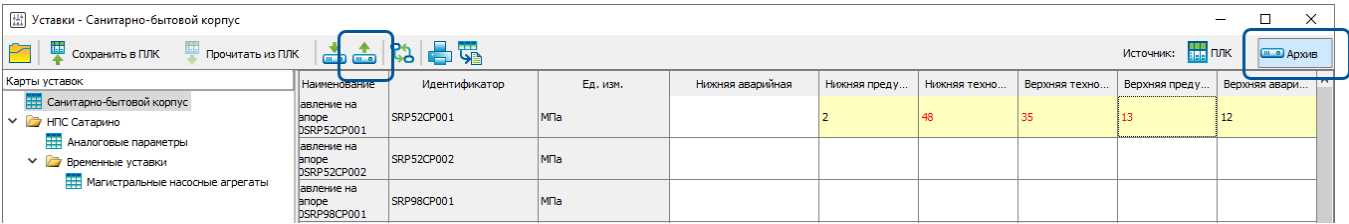
```

Чтобы сохранить значения уставок в архив, перейдите к нужному источнику (ПЛК или Архив) и нажмите кнопку **Сохранить в архив** на панели инструментов. В архив сохраняются именно те значения, которые вы видите для каждой карты уставок в выбранном режиме. Из режима **ПЛК** данные сохраняются в оперативный архив, который тут же подгружается в таблицу режима **Архив**. Чтобы сохранить отдельный файл со значениями уставок, нажимайте кнопку **Сохранить в архив** в режиме **Архив**.



Карты уставок	Назначение	Идентификатор	Ед. изм.	Нижняя аварийная	Нижняя преду...	Нижняя техно...	Верхняя техно...	Верхняя преду...	Верхняя авари...
Санитарно-бытовой корпус									
НПС Сатарино									
Аналоговые параметры									
Временные уставки									
Магистральные насосные агрегаты									
	авление на входе DSRP52CP001	SRP52CP001	МПа		2	48	35	13	12
	авление на входе DSRP52CP002	SRP52CP002	МПа						
	авление на входе DSRP98CP001	SRP98CP001	МПа						

Чтобы загрузить архивный файл, переключитесь на источник **Архив** и нажмите кнопку **Загрузить из архива**. При открытии архивного файла в таблице SePlatform.HMI.SetPoints изменяются значения только тех уставок, которые указаны в файле.



Если архивное значение отличается от значения в ПЛК, ячейка окрашивается в желтый цвет.

№	Наименование	Идентификатор	Ед. изм.	Нижняя аварийная	Нижняя преду...	Нижняя техно...	Верхняя техно...	Верхняя преду...
1	Давление на напоре 00SRP52CP001	SRP52CP001	МПа		1	12	32	10
2	Давление на напоре 00SRP52CP002	SRP52CP002	МПа					
3	Давление на напоре 00SRP98CP001	SRP98CP001	МПа					

История изменений

2.1

Новые возможности

- Добавлена возможность указывать в конфигурационном файле необходимость переноса слов. Чтобы включить/отключить перенос слов, используйте флаг **WordWrap** («true» / «false») у элемента **Columns**.
- Добавлена возможность вызова экранной клавиатуры в диалоговых окнах:
 - задания имени папки для экспорта таблицы в Excel;
 - авторизации;
 - открытия файлов;
 - ввода значений уставок.

Для отображения кнопки вызова клавиатуры нужно включить эту функцию в параметре **init_ScreenKeyboard** (по умолчанию кнопка появляется только в диалоге авторизации).

- Для SePlatform.HMI.SetPoints, работающего в виде отдельного приложения, реализована возможность загрузки настроек из файла `app_settings.json`.

Загрузить можно:

- Параметры работы с виртуальной клавиатурой. Чтобы включить клавиатуру, используйте параметр **Enabled** элемента **ScreenKeyboard**.
- Разрешение использования сервера безопасности для ограничения прав пользователя. Чтобы разрешить использовать сервер безопасности, используйте параметр **UseSecurity** в элементе **Permissions**.

Файл хранится по пути:

- OC Windows: `C:\ProgramData\SePlatform\HMI.SetPoints\`
- OC Linux: `/home/<user>/SePlatform/HMI.SetPoints/`

Улучшение

Теперь текстовая подсказка к ошибке всегда принудительно отображается рядом с полем ввода, содержащим ошибку.

2.1.1

Новые возможности

Улучшен экспорт карт уставок в MS Excel:

- Теперь можно экспортировать не только все карты уставок сразу, но и всего одну - текущую.
- Добавлено диалоговое окно экспорта, в котором можно:
 - Выбрать папку для экспорта.
 - Выбрать тип файла: *.csv или *.xlsx.
 - Ввести произвольный текст, который будет отображаться в верхнем колонтитуле (например, наименование объекта).
 - Указать имя экспортируемого файла.
- Прогресс выполнения экспорта отображается в статусной строке.

Изменение

Настройки колонтитулов при экспорте и печати для всех сразу карт уставок указываются теперь в одном месте - поле **PrintSettings**.

Исправленная ошибка

Устранена причина, по которой в файле оперативного архива отсутствовала запись о флаге уставки.

Изменения документации

Редакция 1

Внутренние изменения. Содержимое документа не менялось.

Редакция 2

Внутренние изменения. Содержимое документа не менялось.

2.0

Новые возможности

- Добавлена возможность подключения к серверу, защищенному паролем.
- Добавлена фильтрация в дереве карт уставок.
- Переработан механизм безопасности. Для ограничения прав пользователей добавлен новый тип Разрешения (**SetPoints_Permissions**). Вместо свойств **Доступность файловой системы** (**FileSystemAccess**) и **Только для чтения** (**IsReadOnly**) типа **SetPoints** теперь используйте свойства **Доступность файловой системы** (**FileSystemAccess**) и **Изменение значений** (**Modification**) типа **Разрешения**. Также новый тип содержит свойства **Выбор принтера** (**PrinterSelecting**), **Разрешенные принтеры** (**AllowedPrinters**) и **Экспорт в файл** (**Export**).

В свойстве **Разрешенные принтеры** (**AllowedPrinters**) принтеры указывайте в виде JSON массива.
["Send To OneNote 2013", "Snagit 2019", "HP LaserJet Pro MFP M426f-M427f PCL 6"]



ОБРАТИТЕ ВНИМАНИЕ

При переходе на SePlatform.HMI.SetPoints 2.0.0 для использования нового типа **Разрешения** вам потребуется доработать проект.

В экземпляре типа **SetPoints** в параметре инициализации **Разрешения** сошлитесь на нужный экземпляр типа **Разрешения**.

Если вы желаете использовать систему безопасности для ограничения прав пользователя:

1. Создайте в подсистеме безопасности приложение для SePlatform.HMI.SetPoints с правами **Modification** (Изменение значений), **FileSystemAccess** (Доступ к файловой системе), **PrinterSelecting** (Выбор принтера), **Printers** (Принтеры) и **Export** (Экспорт в файл). В праве **Printers** (Принтеры) список принтеров нужно указывать отдельными значениями.
2. Для экземпляра типа **Разрешения**:
 - 2.1. Включите использование подсистемы безопасности (в свойстве **Использовать сервер безопасности для определения прав** (**UseSecurity**)).
 - 2.2. Укажите имя приложения, которое вы используете для SePlatform.HMI.SetPoints в подсистеме безопасности (в свойстве **Имя приложения с правами безопасности** (**SecurityApplication**)).

2.0.1

Исправленные ошибки

- Исправлена ошибка, из-за которой в журнал не выводилось сообщение об успешной записи значения на сервер.
- Устранена причина, по которой в таблицу не подгружалась часть уставок из карты уставок.
- Теперь при подключении к серверу, на котором не установлен пароль, больше не появляются беспричинные ошибки и предупреждения.

Изменения документации

Редакция 1

Обновлены скриншоты в разделе с описанием встраивания SePlatform.HMI.SetPoints в проект ([стр. 22](#)).

1.2

Новые возможности

- Добавлен выпадающий список ранее открываемых конфигурационных файлов, из которого можно быстро выбрать и загрузить конфигурацию.
- Реализована возможность подключения к произвольному серверу. Для этого используйте кнопку "Подключиться к серверу" на панели инструментов.
- Добавлены готовые формы для использования в пользовательских проектах:
 - Форма **SetPoints_Form**, которую можно открывать как диалоговое окно. Форма содержит панель инструментов, дерево навигации, таблицу с уставками и строку статуса. Для формы нужно указать инициализирующие параметры: **init_ApSource** ([Ссылка Источник данных](#)), **init_ConfigFilePath** ([Путь к файлу конфигурации](#)), **init_IsReadOnly** ([Только для чтения](#)) и **init_RestoreWindow** - ([Восстанавливать размер и положение окна при запуске](#)).
 - Форма **SetPoints_Page**, которую можно открывать как содержимое фрейма. Форма содержит панель инструментов, дерево навигации и таблицу с уставками (без строки статуса). Для формы нужно указать инициализирующие параметры: **init_ApSource** ([Ссылка Источник данных](#)), **init_ConfigFilePath** ([Путь к файлу конфигурации](#)) и **init_IsReadOnly** ([Только для чтения](#)).

Исправленные ошибки

- Исправлена ошибка, из-за которой не изменялось значение уставки при вводе нового значения с клавиатуры.
- Решена проблема, по причине которой не применялись единицы измерения, указываемые для столбца в конфигурационном файле.
- Исправлена ошибка, из-за которой настройки, записанные в ПЛК в режиме АРХИВ, не отображались в режиме ПЛК.

1.2.1

Изменение

В диалоге редактирования уставки теперь отображается то же самое значение, что и в таблице.

1.2.2

Новые возможности

- Окно редактирования уставки можно вызывать теперь не только двойным щелчком, но и клавишей «Enter».
- Появилась возможность изменять значения сразу нескольких уставок, если предварительно выделить их.
- При отображении чисел теперь используются региональные настройки разделителя целой и дробной части.
- На лист печати добавлены верхний и нижний колонтитулы. В верхнем колонтитуле можно размещать произвольный текст, в нижнем выводится название карты уставок.

Улучшение

Заголовки столбцов теперь автоматически переносятся на новую строку, если не вмещаются в ширину столбца.

1.2.3

Исправленная ошибка

Не отображалось целое число в редакторе значения уставки, если в региональных настройках ОС использовалась группировка по разрядам.

Изменения документации

Редакция 1

- Обновлен скриншот с командами настройки уставки.
- Сказано, что для загрузки конфигурации при запуске необходимо задействовать функцию Open экземпляра типа **SetPoints** ([стр. 26](#)).
- В описании конфигурационного файла:
 - Изменено максимальное число тегов для чтения/записи настроек уставок ([стр. 19](#)).
 - Приведены правила вычисления числа тегов ([стр. 19](#)) в одном конфигурационном файле.
 - Добавлено примечание о логике записи бита ([стр. 18](#)) при включении уставки/сообщения/звука.
 - Указаны типы сигналов для хранения настроек уставки ([стр. 19](#)).

Редакция 2

Внутренние изменения. Содержимое документа не менялось.